CuriosAI at SemEval-2026 Task 8: Hybrid retrieval system with repeated sampling for generation

Aiswariya Manoj Kumar[1], Hiroki Takushima[1], Fumika Beppu[1], Yuki Shibata[1], Daichi Yamaga[1], Takayuki Hori[1],

[1]SoftBank Corp.,

Correspondence: aiswariya.manojkumar@g.softbank.co.jp

Abstract

SemEval-2026 Task 8 (MTRAGEval) evaluates multi-turn Retrieval-Augmented Generation (RAG) under conversational challenges such as non-standalone turns, underspecification, and answerability detection. These conditions amplify retrieval and generation errors that standard single-turn RAG pipelines fail to address effectively. We present a robustness-oriented multi-turn RAG system combining contextual query rewriting, heterogeneous hybrid retrieval fused with Reciprocal Rank Fusion (RRF), domain-adaptive Low-Rank Adaptation (LoRA) reranking, and repeated sampling with metric-guided selection. On the official test set, our approach outperforms the organizers' baselines across all subtasks: Retrieval (nDCG@5: 0.5396 vs. 0.4795), Generation (0.7571 vs. 0.6390), and RAG (0.5486 vs. 0.5366). Our system ranks 5th in Subtask A, 5th in Subtask B, and 7th in Subtask C on the official leaderboard. These results demonstrate that calibrated hybrid retrieval combined with robust generation selection is effective for multi-turn RAG.

1 Introduction

Multi-turn Retrieval-Augmented Generation (RAG) must interpret dialogue context, resolve implicit references, and decide when available evidence is insufficient. MTRAGEval (Katsis et al., 2025; Rosenthal et al., 2026b) targets these behaviors through non-standalone and underspecified queries, as well as answerability-sensitive turns. In such settings, small upstream failures (e.g., imperfect rewriting or retrieval noise) can cascade into ungrounded or mismatched responses, making pipeline robustness central.

Our system targets these failure modes with four components: (i) contextual query rewriting to produce retrieval-ready standalone queries (Zhou et al., 2023; Sun et al., 2023); (ii) heterogeneous hybrid retrieval combining sparse and dense retrievers and fusing candidates via Reciprocal Rank Fusion (RRF) (Cormack et al., 2009); (iii) domain-adaptive reranking using a Low-Rank Adaptation (LoRA) (Hu et al., 2022) fine-tuned Qwen3-Reranker-8B (Zhang et al., 2025); and (iv) repeated sampling with metric-guided selection to reduce generation variance under noisy evidence.

On the official evaluation, we achieve 0.5396 nDCG@5 for Retrieval, 0.7571 for Generation, and 0.5486 for RAG, improving over the official baselines across all subtasks. These results indicate that hybrid retrieval substantially improves coverage in multi-turn settings, while repeated generation with metric-aligned selection enhances stability under noisy retrieval. Remaining challenges include handling deeply underspecified queries and multi-hop reasoning across distant passages.

2 Background

2.1 Task Setup

MTRAGEval evaluates Multi-turn RAG systems under conversational settings that resemble real-world information-seeking dialogues. Each instance is defined as the full dialogue history up to the current turn plus the latest user query. Systems must use conversational context for reference resolution and answerability decisions (Rosenthal et al., 2026b). The benchmark includes three subtasks:

- **Subtask A (Retrieval):** Retrieve relevant passages for the current turn.
- **Subtask B (Generation):** Generate an answer for the current turn using reference passages provided by the organizers.
- **Subtask C (RAG):** Perform end-to-end retrieval for the current turn, followed by grounded generation.

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 1106–1115

Table 1 summarizes the input and output requirements for each subtask.

2.2 Dataset

MTRAGEval builds upon the MTRAG benchmark (Katsis et al., 2025) and extends it with evaluation tasks targeting challenging conversational properties. The development data consists of 110 manually created and reviewed English conversations, comprising 842 tasks across four domains: ClapNQ, FiQA, Govt, and Cloud. We provide additional corpus analysis in Appendix A.

For the test phase, the organizers provided 507 tasks derived from unseen dialogue contexts (Rosenthal et al., 2026b). These test tasks (MTRAG-UN) (Rosenthal et al., 2026a) contain a higher proportion of non-standalone, underspecified, and answerability-sensitive instances compared to earlier benchmark of MTRAG.

2.3 Related Work

Contextual query rewriting has been shown to improve retrieval quality for multi-turn dialogue systems by expanding non-standalone user queries into standalone forms (Zhou et al., 2023; Sun et al., 2023). Hybrid retrieval methods that combine sparse and dense representations, such as RRF, have demonstrated robustness across heterogeneous corpora (Cormack et al., 2009). Adaptation techniques like LoRA enable efficient fine-tuning of large reranking models (Hu et al., 2022), while repeated sampling strategies have been explored to enhance generation reliability (Wang et al., 2023). Our work builds on these foundations and focuses on robustness across stages in multi-turn conversational RAG.

3 System Overview

Our system is designed to mitigate error propagation in multi-turn RAG by explicitly addressing conversational ambiguity, retrieval instability, and generation variance. These are handled through conversational rewriting, hybrid retrieval with domain-adaptive reranking, and metric-guided generation. Figure 1 and 2 illustrate the retrieval and generation pipelines respectively.

3.1 Conversational Query Rewriting

We adopt the contextual query rewriting strategy provided in the official baseline (Katsis et al., 2025). Given the full conversation history and the current user query, GPT-5 (OpenAI, 2026a) rewrites the latest turn into a standalone query that preserves user intent while resolving implicit references.

The rewriting prompt follows the baseline formulation, and differs only in the underlying language model used for generation. This approach is consistent with prior contextual query rewriting work (Zhou et al., 2023; Sun et al., 2023).

3.2 Hybrid Retrieval Framework

3.2.1 Preprocessing and Indexing

We used the organizers' passage-level corpora and retain the original chunking. Before indexing, passages are normalized with NFKC and cleaned to remove control characters and common crawl artifacts (HTML remnants, pagination stubs, and boilerplate).

3.2.2 Summary augmentation

For each passage chunk, we generated a concise summary using Qwen3VL-32B-Instruct (Bai et al., 2025) and appended it as an additional field (`summary`). The original text content remained unchanged. This augmentation enriched dense semantic representations while preserving passage granularity.

3.2.3 Multi-Index Retrieval

We constructed four independent retrieval indices over the cleaned corpora with appended summaries.

- Sparse lexical retrieval: SPLADE-v3 (Lassance et al., 2024)
- Dense embedding: NV-Embed-v2 (Lee et al., 2024)
- Dense embedding: Qwen3-Embedding-8B (Zhang et al., 2025)
- Dense embedding: text-embedding-3-large (OpenAI, 2026b)

Each retriever returns the top-100 passages for the rewritten query. The diversity across sparse and dense representations improves coverage within domain-specific corpora that contain varying lexical and structural characteristics.

3.2.4 Reranking

We fine-tune Qwen3-Reranker-8B with LoRA (Hu et al., 2022) and rerank retrieved candidates. Development results show consistent gains in Cloud but limited or negative impact in ClapNQ, FiQA,

Subtask	Input	Output
Subtask A	Full conversation	Top-10 ranked passages
Subtask B	Full conversation + reference passages	Grounded answer to the last turn of conversation
Subtask C	Full conversation	Grounded answer to the last turn of conversation

Table 1: Summary of inputs and outputs for each MTRAGEval subtask.

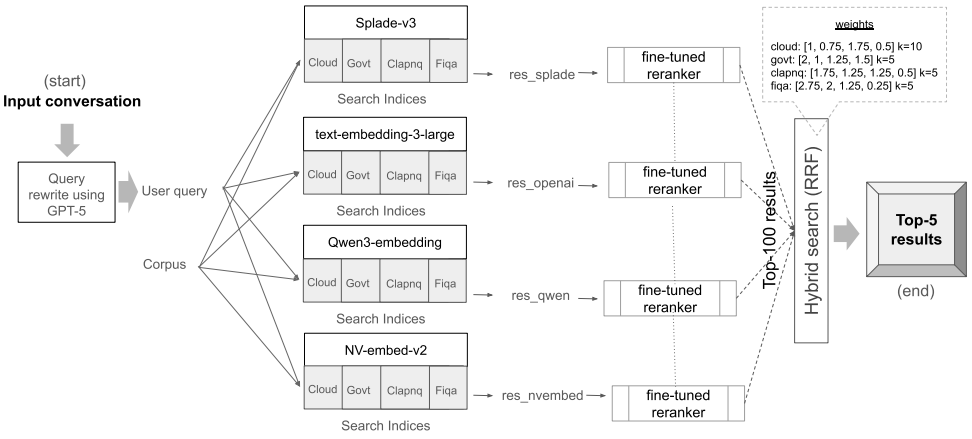

Figure 1: Hybrid retrieval pipeline. The conversational query is first rewritten into a standalone form. Multiple sparse and dense retrievers independently retrieve top candidates, which are fused using RRF and reranked via LoRA-fine-tuned Qwen3-Reranker-8B. The final top-5 passages are passed to the generator.

and Govt (Table 4). We therefore adopt a domain-adaptive strategy: reranking is applied only for Cloud and retains original rankings elsewhere. Additional fine-tuning details are provided in the Appendix B.2.

3.2.5 Weighted Reciprocal Rank Fusion

The reranked lists from all retrievers are combined using a weighted variant of RRF (Cormack et al., 2009). For document d, the fused score is computed as:

$$RRF(d) = \sum_{r \in R} w_r \cdot \frac{1}{k + \text{rank}_r(d)}$$

where R denotes the set of reranked retrieval systems, w_r is the weight assigned to system r, and k is a smoothing parameter.

Weights and the smoothing parameter k were selected via grid search on the development split (details in Section B.1). The final top-10 passages after fusion are provided to the generator.

3.3 Generation Strategy

3.3.1 Prompting

Generation uses GPT-5 with a structured grounding prompt enforcing: (i) reliance solely on retrieved passages for factual content, (ii) explicit handling of fully, partially, and unanswerable queries, (iii) message-type-aware response formatting, and (iv) answering only the latest turn. This aligns generation behavior with official faithfulness and answerability criteria.

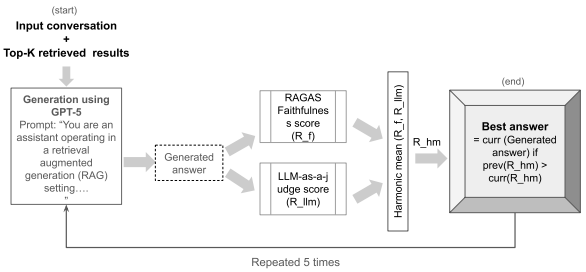

Figure 2: Generation pipeline with repeated sampling and dual-score selection. Five candidate responses are generated and scored using metric-aligned evaluation signals. The final output maximizes harmonic scoring.

3.3.2 Repeated Sampling and Metric-Guided Selection

To mitigate generation variance under imperfect retrieval, we generate $n = 5$ independent candidate responses $\{y_1, \ldots, y_n\}$. This approach increases

the likelihood that at least one candidate is both well-grounded and contextually appropriate. Each candidate is scored using:

- **Faithfulness Score** R_f computed using Retrieval-Augmented Generation Assessment Score (RAGAS) (Es et al., 2025), measuring grounding with respect to the retrieved passages.
- **LLM-as-a-Judge Score** (Zheng et al., 2023) R_llm computed using a GPT-based evaluator with a simplified prompt inspired by the official evaluation setup,[1] focusing on context relevance and hallucination penalization. This internal judge was used only for candidate selection and was not identical to the organizers' official RB_{llm} scorer, since it used an independently written prompt without reference-answer conditioning. The complete prompt is provided in Appendix D.2 for transparency.

The final output is selected as:

$$y^* = \arg\max_{y_i} \ \mathrm{HM}\big(R\_f, R\_llm\big)$$

where HM denotes the harmonic mean. This formulation favors responses that are simultaneously well-grounded and conversationally appropriate, discouraging candidates that score highly on only one criterion.

4 Experimental Setup

4.1 Data Splits

We tune all hyperparameters on the official development split only and report final scores from organizer evaluation on the held-out test set.

4.2 Hyperparameter Tuning

Retrieval hyperparameters for weighted RRF were tuned via grid search on the development split, optimizing retriever weights and smoothing parameter k with respect to the official evaluation metric nDCG@5.

For generation, five candidate responses were sampled per query. No decoding temperature was manually tuned due to API constraints. Final answer selection used metric-guided harmonic scoring.

Detailed search spaces and training configurations are provided in the Appendix B.1.

[1] https://github.com/IBM/mt-rag-benchmark/blob/main/scripts/evaluation/judge_utils.py

Subtask	Our Score	Baseline	Rank (Ours/Total)
Subtask A	0.5396	0.4795	5/38
Subtask B	0.7571	0.639	5/26
Subtask C	0.5486	0.5366	7/29

Table 2: Official test results compared against the organizers' baseline. Rank indicates the position of our system on the official leaderboard for each subtask. Subtask A is evaluated using nDCG@5, while Subtasks B and C use the harmonic mean of RB_{alg}, RL_F, and RB_{llm}. See Appendix B.3 for details.

5 Results

Table 2 reports official test-set performance on all three subtasks, along with our leaderboard rankings. The organizers' baseline follows the MTRAG-UN pipeline (Rosenthal et al., 2026a): (i) contextual query rewriting with gpt-oss-20b, (ii) ELSER-based sparse retrieval for Subtask A, and (iii) structured grounded generation with answerability logic using gpt-oss-120b (Subtask B) and qwen-30b-a3b-thinking (Subtask C). This baseline constitutes a strong multi-turn RAG system under the benchmark's conversational phenomena.

Beyond the organizers' baseline, leaderboard rankings provide broader context, with our system placing **5th** in Subtask A, **5th** in Subtask B, and **7th** in Subtask C, demonstrating competitive performance among participating systems.

Our system improves over the baseline across all subtasks. The largest absolute gain is on **Subtask A** (+0.0601 nDCG@5), suggesting that heterogeneous hybrid retrieval with weighted RRF substantially improves evidence coverage in multi-domain, later-turn settings. We also observe improvements on **Subtask B** (+0.1181) and **Subtask C** (+0.0120), indicating that repeated sampling with metric-guided selection improves generation quality both when passages are fixed (Subtask B) and when retrieval noise is present (Subtask C), though the end-to-end setting remains bottlenecked by retrieval imperfections.

Retrieval results Hybrid retrieval yields a substantial improvement over the baseline sparse-only configuration. In multi-turn settings, rewriting errors and domain heterogeneity often reduce lexical overlap between query and evidence; combining sparse and multiple dense retrievers mitigates this mismatch by improving recall across paraphrases and technical variants. Weighted RRF further stabilizes rankings by leveraging complementary signals from individual retrievers.

Configuration	Average nDCG@5
Raw corpus	0.3293
+ Cleaning	0.3332
+ Summary augmentation	0.3425
+ Query prefixing	0.3921
+ Reranking	0.424
+ Fine-tuned reranking	**0.443**

Table 3: Incremental retrieval improvements on the development split using Qwen3-8B-Embedding index. Each row cumulatively adds the listed modification.

Generation results Our generation improvements are more pronounced in Subtask B than in Subtask C. Because Subtask B provides reference passages, gains primarily reflect improvements in generation robustness—specifically grounding behavior, answerability handling, and response appropriateness. In contrast, Subtask C remains constrained by upstream retrieval quality, limiting the magnitude of end-to-end gains.

The consistent improvement in Subtask B suggests that repeated sampling and metric-guided selection improves stability in conversational settings, particularly for underspecified or answerability-sensitive turns. In Subtask C, improvements are more modest but indicate that the generation strategy remains robust even under imperfect retrieval.

5.1 Ablation Study of Retrieval Refinements

We quantify the impact of retrieval refinements on the development split by progressively adding: (i) corpus cleaning, (ii) passage-level summary augmentation, (iii) model-specific query prefixing for embedding retrievers, and (iv) reranking with a LoRA-tuned reranker.

Table 3 reports development nDCG@5 under these incremental modifications. Each component yielded measurable improvements. Corpus cleaning reduced lexical noise and improved stability across domains. Summary augmentation strengthened dense semantic representations. Query prefixing improved embedding alignment for instruction-tuned models. Fine-tuned reranking further improved ranking quality in certain domains.

5.2 Domain Effects of Reranking

As shown in Table 4, we observed that reranking improves performance in Cloud but degrades performance in FiQA and Govt and has negligible effect on ClapNQ. This motivates our domain-adaptive strategy, applying reranking only for Cloud.

Domain	Pre-reranked RRF	Post-reranked RRF
FiQA	0.4527	0.4191
Govt	0.5114	0.4792
ClapNQ	0.5420	0.5417
Cloud	**0.4188**	**0.4421**

Table 4: Development nDCG@5 comparison of RRF using original ranked lists vs reranked lists on the development dataset.

We attribute this behavior to two factors. First, weighted RRF already produces strong first-stage rankings by integrating sparse and dense signals, leaving limited headroom for reranking in domains where lexical and semantic alignment is already high. Second, corpus characteristics differ substantially across domains (Appendix A). Cloud contains longer, more repetitive technical documentation (lower lexical diversity), where semantic reranking can better separate structurally similar passages. In contrast, FiQA and ClapNQ contain shorter, more lexically diverse passages, and Govt contains substantial noise; in these cases, reranking can amplify spurious semantic matches or reduce robustness to boilerplate.

5.3 Error Analysis

Manual inspection reveals two recurring issues. First, underspecified queries that implicitly required clarification were typically answered using available evidence rather than prompting for clarification. Second, generated responses were often more verbose than ground-truth references, which may negatively affect RB_{alg} despite factual correctness. Introducing explicit length control could improve alignment with reference-style answers.

Most errors were attributable to retrieval noise rather than clear hallucination, though we did not conduct a dedicated hallucination audit.

6 Conclusion

We presented a robustness-oriented multi-turn RAG system for SemEval-2026 Task 8 (MTRAGEval) integrating conversational query rewriting, hybrid retrieval with weighted RRF, domain-adaptive reranking, and metric-guided repeated generation. The system consistently outperforms official baselines across all subtasks, highlighting the importance of mitigating error propagation in conversational pipelines.

Future work includes feedback-driven retrieval refinement, task-specific generation fine-tuning, and improved structured retrieval strategies.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 758–759.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2025. Ragas: Automated evaluation of retrieval augmented generation. *Preprint*, arXiv:2309.15217.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*.

Yannis Katsis, Sara Rosenthal, Kshitij Fadnis, Chulaka Gunasekara, Young-Suk Lee, Lucian Popa, Vraj Shah, Huaiyu Zhu, Danish Contractor, and Marina Danilevsky. 2025. mtrag: A multi-turn conversational benchmark for evaluating retrieval-augmented generation systems. *Transactions of the Association for Computational Linguistics*, 13:784–808.

Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. 2024. Splade-v3: New baselines for splade. *Preprint*, arXiv:2403.06789.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *Preprint*, arXiv:2405.17428.

OpenAI. 2026a. Gpt-5 api documentation. `https://developers.openai.com/api/docs/models/gpt-5`.

OpenAI. 2026b. text-embedding-3-large api documentation. `https://developers.openai.com/api/docs/models/text-embedding-3-large`.

Sara Rosenthal, Yannis Katsis, Vraj Shah, Lihong He, Lucian Popa, and Marina Danilevsky. 2026a. Mtrag-un: A benchmark for open challenges in multi-turn rag conversations. *Preprint*, arXiv:2602.23184.

Sara Rosenthal, Vraj Shah, Yannis Katsis, and Marina Danilevsky. 2026b. Semeval-2026 task 8: Mtrageval: Evaluating multi-turn rag conversations. In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*, San Diego, California. Association for Computational Linguistics.

Zhongkai Sun, Yingxue Zhou, Jie Hao, Xing Fan, Yanbin Lu, Chengyuan Ma, Wei (Sawyer) Shen, and Chenlei (Edward) Guo. 2023. Improving contextual query rewrite for conversational ai agents through user-preference feedback learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23.

Yingxue Zhou, Jie Hao, Mukund Rungta, Yang Liu, Eunah Cho, Xing Fan, Yanbin Lu, Vishal Vasudevan, Kellen Gillespie, and Zeynab Raeesy. 2023. Unified contextual query rewriting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 608–615.

A Corpus Exploratory Data Analysis

We conducted exploratory data analysis on the passage-level corpora provided by the organizers to better understand domain-specific characteristics and potential retrieval challenges.

A.1 Qualitative analysis

A.1.1 General Observations

All corpora are provided at the passage level, with relevance judgments defined over passage identifiers. Each passage contains metadata fields such as title and/or URL depending on the domain. Titles, when present, are prefixed to the passage text.

A.1.2 ClapNQ (Wikipedia-based QA)

The ClapNQ corpus contains titles but no valid URLs. Content is largely educational and scientific, covering topics such as normal distributions,

circular motion, Planck's law, Lagrangian mechanics, radioactive decay, and electromagnetic radiation. Many passages include mathematical expressions, Unicode characters (e.g., Greek symbols), and multilingual text fragments (e.g., Japanese, Arabic, Hindi, Hebrew, Cherokee scripts). Some passages contain flattened tables (e.g., racing results) and formula-heavy text, which may affect lexical matching.

A.1.3 FiQA (financial advice from StackExchange)

The FiQA corpus does not include URLs or titles and consists of relatively short passages derived from financial discussion forums. The language is conversational and informal, frequently containing slang, profanity (e.g., abbreviated or partially masked terms), first-person perspectives, and opinionated statements. We observed occasional empty passages and sparse mathematical expressions. The informal style introduces variability that may impact both sparse and dense retrieval behavior.

A.1.4 Cloud (technical documentation from cloud services)

The Cloud corpus includes valid URLs but no titles. Content primarily consists of cloud CLI documentation and technical instructions. Passages contain numerous newline characters, hyphenated command flags, unstructured lists, and in some cases numeric-only segments likely derived from chart data. Repeated artifacts such as SVG icon URLs and template markers were observed, necessitating targeted cleaning.

A.1.5 Govt (web-crawled content from selected .gov and .mil domains)

The Govt corpus contains both valid URLs and titles and is the noisiest among the four domains. Content includes structured court records, privacy policies, and administrative documents. We observed extensive HTML remnants, pagination markers (e.g., repeated page indicators), long web archive URLs, and multilingual Unicode text (including Korean, Hebrew, Vietnamese, Urdu, and Russian scripts). These artifacts increase lexical noise and motivated more aggressive preprocessing.

A.1.6 Implications for Retrieval

The four domains exhibit substantial heterogeneity in writing style, structure, and noise patterns. ClapNQ is formula-heavy and multilingual, FiQA is informal and conversational, Cloud is technical and semi-structured, and Govt contains significant web crawl artifacts. These differences motivated our hybrid retrieval design combining sparse and dense representations, domain-adaptive reranking, and preprocessing steps to mitigate noise.

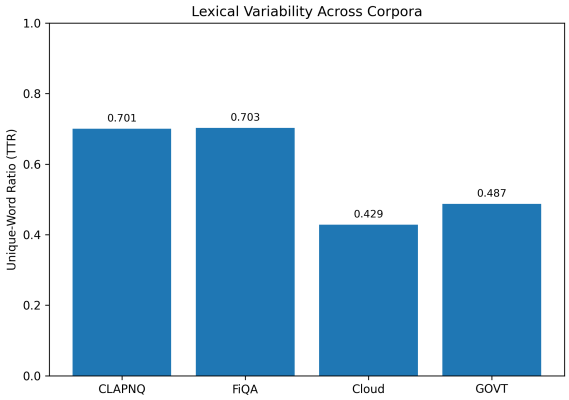

Figure 3: Lexical variability (TTR) across domains. Cloud and Govt exhibit lower diversity than ClapNQ and FiQA.

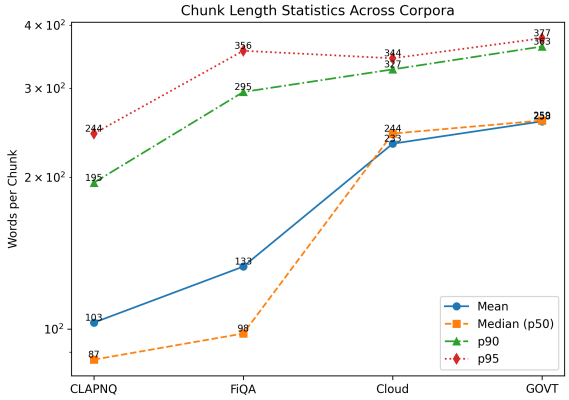

Figure 4: Chunk length (word count) statistics across domains. Cloud and Govt passages are approximately twice as long as ClapNQ, reflecting longer technical and administrative document structure.

A.2 Quantitative analysis

We measure lexical diversity using the Type–Token Ratio (TTR), defined as the ratio of unique tokens to total tokens in a passage. Cloud and Govt passages are substantially longer (mean 233 and 258 tokens respectively) compared to ClapNQ (103 tokens) and FiQA (133 tokens). Lexical diversity, measured via unique-word ratio (TTR), is considerably lower in Cloud (0.43) and Govt (0.49) than in ClapNQ and FiQA (≈ 0.70).

These results indicate that Cloud contains longer and more lexically repetitive technical documenta-

tion, where semantic reranking provides stronger disambiguation. In contrast, ClapNQ and FiQA contain shorter, lexically diverse passages where hybrid retrieval already achieves strong alignment.

B Additional hyperparameters

B.1 Retrieval

Retrieval hyperparameters for RRF were tuned using grid search on the development split to select optimal retriever weights and the smoothing parameter k, targeting nDCG@5. The smoothing parameter was searched over $k \in 5, 10$.

Retriever-specific weights were searched over the following ranges:

- SPLADE-v3 (w_S): $[0.0, 3.0]$ in increments of 0.25
- Qwen3-Embedding-8B (w_Q): $[0.0, 2.0]$ in increments of 0.25
- NV-Embed-v2 (w_N): $[0.0, 2.0]$ in increments of 0.25
- text-embedding-3-large (w_O): $[0.0, 3.0]$ in increments of 0.25

Weights for SPLADE-v3 and text-embedding-3-large embeddings were searched over a larger range based on preliminary experiments indicating stronger standalone retrieval performance relative to Qwen3-Embedding-8B and NV-Embed-v2 embeddings.

Grid search was conducted separately for (i) fused ranked lists prior to reranking and (ii) reranked outputs. Due to computational constraints, we did not jointly optimize fusion weights across both stages.

B.2 Reranking

For reranker training (Qwen3-Reranker-8B), we constructed 1,665 query-document supervision instances from rewritten queries, annotated positive contexts, and SPLADE-v3 sparse retrieval results. Positive chunks were taken from ground-truth contexts, while hard negatives were mined after excluding positive chunk IDs using rank-stratified sampling:

- 2 from top 10
- 3 from ranks 11–50
- 3 from ranks 51–100

During training, we capped each instance to at most 1 positive and 6 negatives, yielding multiple positive–negative pairs per query and effectively increasing supervision. We formulate reranking as a pointwise binary classification problem, where each query–chunk pair is independently labeled as relevant or non-relevant, and optimize a binary cross-entropy loss. We held out 16% of the data for validation and employed LoRA for fine-tuning, with hyperparameters summarized in Table 5.

Parameter	Value
Learning rate	6×10^{-6}
Epochs	3
LoRA rank	8
LoRA α	32
Gradient accumulation	16
Mixed precision	BF16

Table 5: LoRA fine-tuning hyperparameters for Qwen3-Reranker-8B.

B.3 Evaluation Metrics

Subtask A (Retrieval) was evaluated using normalized Discounted Cumulative Gain at rank 5 (nDCG@5).

Subtasks B and C were evaluated using the harmonic mean of three metrics defined by the organizers (Rosenthal et al., 2026b):

- RB_{alg}: harmonic mean of BERTScore Recall, ROUGE-L, and BERT-K Precision,
- RB_{llm}: reference-based LLM judge score,
- RL_F: faithfulness score measuring grounding with respect to retrieved passages.

Evaluation is conditioned on answerability classification via an "I Don't Know" (IDK) judge as described in the task overview papers.

B.4 Implementation Details

Sparse retrieval was implemented using SPLADE-v3. Sparse retrieval, dense retrieval and reranking models were obtained from HuggingFace Transformers.[2] LoRA fine-tuning was implemented using parameter-efficient training utilities.[3] Generation was performed using GPT-5 via the Azure OpenAI API.

[2]https://huggingface.co/transformers
[3]https://github.com/huggingface/peft

Domain	NV-emb	Qwen3-emb	text-emb	SPLADE
FiQA	0.389	0.395	0.407	0.404
Govt	0.453	0.437	0.453	0.465
ClapNQ	0.536	0.533	0.535	0.537
Cloud	0.422	0.407	0.421	0.425

Table 6: Development nDCG@5 of individual retrievers across domains prior to fusion.

C Ablation study details

C.1 Retrieval

To quantify the contribution of individual retrieval refinements, we conducted controlled experiments on the development split, progressively modifying the retrieval pipeline.

We compare the following configurations:

1. Raw corpus indexing
2. Cleaned corpus (Unicode normalization and boilerplate removal)
3. Cleaned corpus with summary augmentation
4. Query prefixing for embedding models
5. Domain-adaptive reranking

Query Prefixing For embedding-based retrieval, we applied model-specific query prefixes to better align query representations with training objectives. For NVIDIA embeddings, we used:

```
Instruct: Given a search query, retrieve
relevant passages that answer the query.
Query: <query>
```

For other embedding models, we applied:

```
Represent this sentence for searching
relevant passages: <query>
```

Query prefixing yielded consistent improvements in dense retrieval performance, suggesting better alignment between query encoding and model pretraining instructions.

C.2 Model performance comparison

Table 6 reports development nDCG@5 scores of individual retrievers prior to fusion. Performance varies across domains, reflecting differences in corpus characteristics.

SPLADE-v3 achieves the strongest performance in Govt and ClapNQ, indicating the effectiveness of sparse lexical matching in domains containing formal or entity-heavy content. In FiQA, dense embedding models slightly outperform SPLADE-v3, likely due to the conversational and semantically varied nature of financial discussion data.

In the Cloud domain, performance differences among retrievers are relatively small, suggesting that no single retrieval paradigm dominates. This further motivates the use of weighted RRF to combine complementary signals.

D Prompt Design

D.1 Generation prompt

Due to space constraints, we summarize the key components of the system prompt used for generation in Subtask B and C rather than reproducing it in full.

The prompt enforces strict grounding and answerability behavior in a RAG setting through the following mechanisms:

Knowledge Restriction. The model is instructed to treat retrieved passages as its entire knowledge base for factual, explanatory, and procedural queries. It is explicitly prohibited from introducing external knowledge or inferring unstated facts.

Answerability Classification. For informational queries, the model must classify the request as fully answerable, partially answerable, or unanswerable based solely on retrieved evidence. Unanswerable cases require a natural refusal (e.g., "I do not have enough information"). Partially answerable cases must clearly separate supported and unsupported content.

Message-Type Conditioning. The prompt includes behavioral rules for different query types (factoid, summarization, explanation, comparative, troubleshooting, conversational). Informational responses must be grounded, while purely social turns may use generic conversational language without introducing new facts.

Multi-Turn Constraints. The model is instructed to answer only the latest user turn, using prior dialogue history solely for contextual grounding (e.g., pronoun resolution).

Metric Alignment. The prompt design explicitly targets faithfulness and answerability, aligning generation behavior with the official evaluation metrics used in MTRAGEval.

D.2 LLM-as-a-Judge prompt

In order to enforce contextually appropriate answers, we used the following prompt to compute

the LLM-as-a-Judge Score.

```
You should act as a judge and evaluate the given answer
      against the user question and provided contexts.

Scoring: Rate the answer strictly on a continuous scale from
      0.0 to 1.0 where:
1.0 -> fully faithful to the document, appropriate to the
      question, and complete
0.0 -> unfaithful, irrelevant, hallucinated, or unsupported
0.25 -> weak, partially hallucinated or incomplete
0.5 -> moderately correct but missing details or partially
      faithful
0.75 -> mostly correct, minor omissions or minor irrelevancies

You MUST output ONLY the final rating number. No explanation,
      no sentences, no extra text. Just the number.
```